# Task-allocation and Coordination of Multiple Robots for Planetary Exploration

Maja J. Matarić*  Gaurav S. Sukhatme

mataric,gaurav@usc.edu
http://robotics.usc.edu
Robotics Research Laboratories
Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0781

## Abstract

*We address the problem of dynamic task allocation and present a review of three multi-robot coordination strategies we have studied to address this problem. Each strategy is tested using embodied multi-robot experiments within a separate problem domain. The three experimental problem domains (object tracking, object manipulation, and emergency-handling) are loosely motivated by future cooperating planetary robot rovers. For ease of comparison, we situate the three approaches within a unified framework, and report experimental results.*

## 1 Introduction

Following the spectacular success of the Sojourner robot rover [14], it is likely that future missions to Mars (and other planets) will rely on mobile robot explorers. In particular, *multiple cooperating robots* hold the promise of improved performance and increased fault tolerance for large-scale problems such as planetary survey and habitat construction [21, 10]. NASA already has tentative plans for multi-robot missions [1] in the coming decades.

Fielding multiple robots for space exploration is a daunting challenge [20], since basic research in the area is still addressing fundamental problems that are currently unsolved in multi-robot system design and coordination. While there has been significant prior research in multi-robot collaboration [2, 11, 4] and task allocation [17], the general problem of *dynamically allocating tasks in a group of multiple robots satisfying multiple goals*, is as yet unsolved.

In this paper, we focus on three studies that address this problem, which were carried out in our labs. We present a unified framework to clarify the relative similarities and differences among the three approaches, as well as to address the problem at a more general level. The first approach we describe is opportunistic (robots grab a task if they think they are eligible to perform it), and uses broadcast of local eligibility and mutual inhibition among robots [22]. The second approach, in contrast involves commitment to the selected task [8], and task allocation is based on a market-based auction system. The third approach [15] studies the tradeoff between opportunism and commitment relative to cooperative vs. mutually exclusive individuals in a multi-robot team.

These approaches share a common underlying theme: each addresses an instance of the general problem of multi-robot task allocation for multiple goals. For purposes of generality, we address three different experimental problem domains that require multi-robot coordination: multiple target tracking, object manipulation, and emergency handling. The target tracking problem requires robots to maintain all visible "targets" within their field of view; this has NASA-relevant applications involving tracking humans or other robots, for maintaining safety. One could imagine a scenario where humans are engaged in an exploration task and the robots are charged with keeping all the humans within view so none gets lost. The second experimental problem domain, object manipulation, is an example of a tightly-coupled coordination problem, where robots must relocate objects larger than themselves by cooperatively pushing. This problem has direct applications for habitat construction

*contact author for correspondence

on a planetary mission, where material transport will play a key role. The third experimental domain, emergency handling, requires robots to collectively handle any problems that arise in their environment, such as gas leaks, structural failures, injuries to humans, etc. In any long-term planetary mission, the role of such maintenance robot teams will be vital.

The rest of this paper is organized as follows. We present a brief survey of related work (Section 2), followed by a discussion of the three approaches mentioned above in the context of a unified framework (Section 3). Next we describe each approach and a summarize the experimental results (Sections 4, 5 and 6). We conclude with a discussion and directions for future research.

## 2 Related Work

Within the NASA community, there has been recent interest in using multiple robots for planetary exploration. In [21] an approach based on closely-coordinated robots is presented for planetary outpost construction. In [20] a coordination and control architecture is described for robots on planetary outposts which will serve as precursors [10] to human exploration of Mars.

There has been significant prior research in multi-robot collaboration, including [2, 11, 4]. The ALLIANCE architecture [17] presents a robust, multi-robot, task allocation system. An opportunistic approach based on mutual inhibition and broadcast of local eligibility is presented in [22]. The approach is applied to a target tracking problem studied previously in [18, 19]. An auction-based approach based on commitment is presented in [8], where a task allocation strategy using a market-based auction system commits the robots to their tasks until success or failure. The latter two approaches are summarized in this paper.

## 3 Unifying Framework

The three approaches to multi-robot coordination we report here can be viewed using the following common framework. Imagine that all communication among the robots is done through a "blackboard" [6]. To simulate the experiments with inter-robot communication we describe later, imagine that each robot sends its relevant state information to the blackboard at a fixed frequency (e.g., 10 Hz), and the blackboard information is read by all the robots at a lower frequency (e.g., 1 Hz).

| *Blackboard* | | Tasks | | | |
|---|---|---|---|---|---|
| Robot | Engagement | A | B | C | D |
| 1 | none | 1400 | 20 | 0 | 0 |
| 2 | A | 800 | 80 | 0 | 0 |
| 3 | B | 900 | 40 | 0 | 0 |

Table 1: *Eligibility and engagement information on the blackboard*

In the case of no inter-robot communication, the blackboard simply contains information from one robot. The information on the blackboard is the current engagement of each robot (the task to which the robot is assigned), and the current task eligibility of each robot, as shown in Table 1. Each robot decides its own eligibility based on simple local criteria discussed later.

The basic idea is that if all the robots have the same blackboard information available to them and run the same task-allocation algorithm, they should all come to the same conclusion as to which robot should pursue which task. This is equivalent to a centralized representation of shared group state of the system.

Our *broadcast of local eligibility (BLE)* approach, applied to the target tracking problem, can be understood in the blackboard context as follows. Each robot broadcasts its eligibility (for tracking a particular target) to the others continuously. The "winner" corresponds to the robot with the maximum eligibility entry in the blackboard. This entry (row and column) are then removed from the blackboard and the process is repeated for successive entries until the task assignment is complete. The entire process (starting with a new table of eligibilities) is repeated at a high frequency.

Our *market-based auction (Murdoch)* approach, applied to the object manipulation problem, can be viewed as a similar strategy. The key difference is that the "winner" is decided at a lower frequency through an auction. In contrast, BLE can be viewed as almost a continuous auction.

Finally, we use the emergency handling problem to compare the impact of commitment vs. opportunism, the key difference between BLE and Murdoch, above. We implement both within the following simple blackboard-based task allocation algorithm:

**Step 1** All robots engaged in a task they cannot sense have their engagement set to 'none'.

**Step 2** In the case of a commitment-based strategy, all entries on the blackboard for robots already engaged in a task are set to zero, along with all entries for all tasks already being pursued. In the case of an opportunism-based strategy, this step is skipped.

**Step 3** The highest non-zero score in the table is found, and the robot corresponding to this entry is assigned to the task corresponding to this entry. Then all entries of this robot and the corresponding task are set to zero. Step 3 is repeated with the new table, until no new assignments are made.

In the next sections we describe in more detail the BLE and Murdoch approaches to distributed multi-robot coordination, and their experimental validation. Then we discuss the comparison between their key properties in the context of the blackboard framework. It is important to note that while we use the blackboard to compare and evaluate the different strategies, both BLE and Murdoch are implemented and validated on fully distributed, local hardware and algorithms, not a centralized blackboard model. Each is implemented using behavior-based control [12, 3].

# 4 Port-Arbitrated Task Allocation

While it has often been hypothesized that there need be no distinction between inter-robot and inter-behavior communication, no previous system has provided standard tools that allow port-based messaging, suppression, and inhibition between behaviors on separate networked and communicating robots. The goal of this work was to demonstrate that behavior-based systems, restricted to well-defined port-arbitrated interactions, can scale to higher levels of competence than is generally assumed. Specifically, we showed that when the port-arbitration paradigm is extended across networks, the resulting systems are able to dynamically reconfigure themselves in order to allocate resources in response to task constraints, environmental conditions, and system resources. We developed Broadcast of Local Eligibility (BLE) as a general tool for coordination between robots.

## 4.1 Broadcast of Local Eligibility

Our Broadcast of Local Eligibility approach investigates the possibilities of extending the *port-arbitrated behavior* (PAB) paradigm across networks of robots. In port-arbitrated behavior-based control (PAB) systems, controllers are written in terms of behaviors, which are groups of concurrent processes that share a public interface. This interface is composed of ports, which are registers that each hold a single data item. Ports in different behaviors are linked together by connections, which are unidirectional data paths between a source port and a destination port. A port can have any number of incoming and outgoing connections. When data is written to a port, either directly from a process within the behavior or indirectly through a connection, they are generally propagated along all of that port's outgoing connections. We say "generally" because data flow can be modified by special connections which may suppress, inhibit, or override data flowing through other connections.

It is through these mechanisms of suppression and inhibition that Subsumption hierarchies [5], as well as other forms of arbitration, can be efficiently and intuitively implemented. Since connections are external to the behaviors, behavior code is easily re-usable, and interaction between behaviors can be modified dynamically. The port abstraction enforces a data-driven approach to programming that "grounds" computation in sensor readings and effector actions. The PAB approach allows a clean, uniform interface between system components (behaviors) at all levels that abstracts away many issues of timing and communication; the "black boxes" of behaviors may contain reactive mappings or even deliberative planners.

Our Broadcast of Local Eligibility (BLE) mechanism is a standard tool comprised of three specific ports added to BLE-arbitrated behaviors – *Local*, *Best*, and *Inhibit*. Each robot makes a local (i.e., derived from data from the robot's own sensors) estimate of its own eligibility for a some task. This eligibility estimate is written to the appropriate behavior's *Local* port, which is connected so as to broadcast this estimate to the *Best* port of each behavior of the same name on every robot on the local network. The *Best* port filters all the incoming messages for the maximum. A comparison is made between the locally determined eligibility (the *Local* port's value) and the best eligibility calculated by a peer behavior on another robot (the *Best* port's value). When a robot's local eligibility is best for some behavior $B_n$ which performs task $T_n$, it writes to its *Inhibit* port, which is connected so as to inhibit the peer behaviors (that is, behaviors $B_n$) on all other robots. In this manner, the most eligible robot "claims" task $T_n$. Since this inhibition is an active process, failure of a robot which has claimed a task results in the task being immediately "freed" for potential takeover by another robot. Since BLE is based on broadcast messages and receiving ports that filter their input for the "best" eligibility, BLE-based systems are inherently scalable. Up to the limit of communication bandwidth,

| Part | Heterogeneous CMOMMT | Heterogeneous W-CMOMMT | Homogeneous CMOMMT | Homogeneous W-CMOMMT |
|---|---|---|---|---|
| 1 | 0.790319 | 0.884551 | 0.815818 | 0.883333 |
| 2 | 0.749194 | 0.833049 | 0.842464 | 0.889321 |
| 3 | 0.737017 | 0.807958 | 0.921923 | 0.932229 |

Table 2: *Averages over 10 heterogeneous / 5 homogeneous trials*

any number of BLE-enabled robots added to a system will properly interact. BLE allows heterogeneous robots to efficiently allocate themselves to appropriate tasks without the need for any explicit communication or global knowledge of particular abilities. The ability to dynamically instantiate and connect BLE-enabled behaviors allows systems to scale in capability as well as in number of robots.

## 4.2 The Problem Domain: Multiple-Target Tracking

We have validated our BLE approach through experiments in the domain of cooperative multi-robot observation of multiple moving targets, or CMOMMT [19]. CMOMMT involves a team of robots which must attempt to keep a number of prioritized moving targets under constant observation. We also implemented a weighted version of the problem, called W-CMOMMT, in which the different targets are priorities. To apply BLE to this problem domain, each robot has behaviors referred to as *Observers*, each of which is parameterized to cause the robot to attempt to stay within observation range of a specific Target (i.e., *Observer1* causes a robot to track *Target1*). A *Search* behavior on each robot causes the robot to wander randomly (intended to be used when no suitable Targets are within the visual field). BLE was used to arbitrate between these behaviors, to determine which task (a specific target or search) each robot in the system should attend to. Results have demonstrated that BLE is able to efficiently assign robots to subtasks in response to differences in robot capabilities and environmental situations, maintaining better coverage of targets than three other arbitration schemes used for comparison.

## 4.3 Experiments & Results

### 4.3.1 Comparison with a Centralized Controller

We compared the performance of BLE with that of a centralized controller. In the centralized controller, a single behavior (running on a desktop computer in our experiments) had the visual information from all the robots and assigns robots to tasks. Average scores for five trials at each cycle time of the centralized controller were:

| Cycle Time | CMOMMT | W-CMOMMT |
|---|---|---|
| 1 Sec | 0.883079 | 0.908655 |
| 40 Sec | 0.830999 | 0.842469 |

Here we see a significant decline in performance from the 1-second cycle case to the 40-second cycle case ($p < 0.0067$ for CMOMMT, $p < 0.015$ for W-CMOMMT), though we do not see a significant difference between 1-second BLE and 1-second Centralized, or between 40-second BLE and 40-second Centralized.

As we would expect, the Centralized controller, with visual information from all robots, performs better, but only marginally so, than the BLE controller at the 1-second cycle time. The BLE controller, however, performs marginally better at the 40-second cycle time, due to its ability to switch between non-cross-inhibited behaviors at will, according to local priorities, regardless of cycle time.

### 4.3.2 Task Spawning and Dynamic Leader Selection

In another set of experiments, we periodically introduced new targets into the environment, and added a robot team "leader" behavior which monitored the environment for such new targets, commanded all robots to instantiate new behaviors to track such new targets, and made "first contact" with them. The leader role had to always be filled, and eligibility for the role was inversely proportional to the priority of the best target observed.

The leader plays a further role: if it finds that the current distribution of robots over the targets is undesirable (e.g., if a target is not observed but a lower-priority target is observed), the leader tracks that target, causing the robot tracking the lower-priority target to take over the leadership task as a result of the eligibility calculation.

| CMOMMT | W-CMOMMT |
|---|---|
| 0.857350 | 0.901249 |

In a typical trial, target coverage is very consistent when possible, although switches of robots covering targets are more frequent than in the non-*Survey*or cases. The target with the lowest priority is the only one to experience significant gaps in coverage. The robots are able to appropriately switch between the *Survey*, and *Wander*, and various *Observer* tasks in order to best cover the prioritized task space. Selection of robots for the "leadership" role of *Survey*or and the ability of the *Survey*or to both reconfigure the team distribution (through "filling the ranks") and controller structure have been shown to be highly effective.

#### 4.3.3 Heterogeneous Robots

We also examined the ability of BLE to effectively assign tasks to heterogeneous robots. Each of the four robots began with a unique set of capabilities, with some overlap.

> Robot $r_1$ can *Observe* targets $t_1$ and $t_2$, and perform *Survey*;
> Robot $r_2$ can *Observe* targets $t_1$ and $t_2$;
> Robot $r_3$ can *Observe* targets $t_3$ and $t_4$, and perform *Survey*; and
> Robot $r_4$ can *Observe* targets $t_3$ and $t_4$.

Average scores over 5 trials were:

| CMOMMT | W-CMOMMT |
|---|---|
| 0.764062 | 0.848296 |

Scores are significantly lower than for the previous experiment with homogeneous robots. To better examine the causes of this difference, we examine the average performance for the separate trial periods. In the heterogeneous case of our CMOMMT experiments, BLE assigns robots to tasks appropriately, though not always optimally. However, unless target motion patterns are known in advance or omnisciently, there is no information available that would allow an optimal algorithm to be constructed. The question of whether a controller (distributed or centralized) which has knowledge of the number of individual robots and their capabilities, and history of targets observed, could generate better task distributions remains open for future study. Details of this work can be found in [24, 23].

# 5 Auction-Based Task Allocation

## 5.1 The Problem Domain: Box-Pushing

The experimental problem domain we address here is cooperative planar manipulation, specifically moving a box, large relative to the size of the robots, from some initial location to some observable (to at least one robot) goal location. We take inspiration from human solutions to analogous problems, and divide the solution into heterogeneous cooperative robots: a watcher and multiple pushers. The "watcher" can see both the current position of the object and the goal, and thus can compute the error signal, perhaps in the form of a correction angle, that can be communicated to the "pushers", usually as a higher-level command, such as "push more on the right."

We formally define this problem with a set of constraints. First, both the box and the goal are observable, and there is an obstacle-free straight-line path between them wide enough for the box and robots to pass; we consider negotiating obstacles only as part of individual low-level control, not in a coordinated fashion. Second, the box is large compared to the robots so that at least two pushers are accommodated. Third, the robots can only move the box by pushing through frictional contact. Fourth, the pushers cannot directly perceive the goal due to the size of the box.

Our system involves two `pushers` and a `watcher`. The `pushers` can see the box, but not the goal, and the `watcher` can see the goal and, while servoing on it, can accurately perceive (using a scanning laser range-finder) the angular error of the box's orientation with respect to the straight-line path from the box to the goal. The goal is to, incrementally and in a coordinated fashion, rotate the box until the angular error is zero, while simultaneously translating it toward the goal. To achieve this, we employ Murdoch, our market-based task allocation system, described next.

## 5.2 Murdoch

Murdoch is a general-purpose task-allocation system we designed for use in dynamic environments in which many robots may come and go at any time. Communication in Murdoch is fundamentally anonymous, through the use of the *publish/subscribe* paradigm: messages are not addressed to individual robots, but rather are tagged with a descriptive *subject* and are *published* onto the network for anyone to hear. A robot registers interest in a particular subject by *subscribing* to it; that robot will receive a published message if the message's subject(s) match the robot's subscription list. The details of Murdoch are given in [7].

Since we are concerned with timely and efficient allocation of resources to tasks, in Murdoch we use subjects to represent resources, which can be physical devices, such as a camera or a microphone. They can also be more abstract representations of a robot's capabilities or current state, such as the possession of a map of the building, or having sufficient energy reserves. Each robot is always subscribed to the set of subjects representing its currently available resources. Thus, to send a message to every robot that has a gripper and camera, we address the message as: {`gripper camera`}. The auction-based task allocation runs on top of this resource-centric addressing scheme, as illustrated in the following example.

We use two robots with cameras, and a third with both a camera and a laser range-finder. The user poses a `relocate-box` task to Murdoch; this task is hierarchical, composed of a `watch-box` task that has two children `push-box` tasks. The `watch-box` task is published as a *task announcement*, and is addressed to {`mobile laser camera`}. The one robot with those resources responds, claiming the task and becoming our `watcher`. The `watcher` begins executing the `watch-box` task, which consists of: finding the goal, determining the angular error of the box, evaluating the control equations, and announcing new pushing tasks. Each `push-box` task is announced to {`mobile camera`}, and is accompanied by a *metric* that pushers can use to evaluate their fitness for the task. Metrics can involve any arbitrary computation and take as input any part of the robot's state; in this case, the metric is a measure of how well-positioned the robot is for pushing on a certain end of the box. For example, when the task is to push on the right end, the metric will reflect whether the box is offset to the left in the robot's camera image. Each candidate executes the metric and publishes its score back to the others, and so everyone immediately knows which robot was the winner (the robots are honest, and tie-breaking mechanisms are built-in). The `watcher`, as auctioneer, awards the winner a time-limited task contract, then enter the monitoring phase. Left and right pushing tasks are allocated in pairs, parameterized with appropriate velocities, based on the orientation of the box. We use time-limited contracts. In our box-pushing domain, each `push-box` task is 3 seconds in length. During those 3 seconds, the `watcher` can, if it sees fit, *renew* either or both pending contracts; alternatively, the contracts can expire, and new, more appropriate tasks can be assigned.

In a typical run, the `watcher` initially announces left and right `push-box` tasks with proper velocities, and lets them push until the box's orientation changes sufficiently to warrant different pushing velocities and thus new tasks. At that point, the current contracts are allowed to expire, and new ones are formed. This reactivity to world conditions is the feature that enables Murdoch to dynamically reassign tasks in the case of failures. For example, when only a single robot is available, the `watcher` will try to allocate two pushing tasks as usual, but only one (the one that has the higher velocity and thus higher priority) will be claimed. That single contract is renewed and the robot pushes on one end of the box until the orientation changes enough that the it is more important to push the other end, at which point the robot will simply switch sides. When another robot is introduced, it will claim the next available pushing task and the two robots will cooperate at pushing the box. We validate this experimentally below.

### 5.3 Experiments & Results

To evaluate the box-pushing system, we performed three sets of experiments on our group of Pioneer robots.[1] We define success as the situation in which the `watcher` declares that the task is terminated, and the center of the box is positioned within 0.25 meters of the target location; we do not specify a target orientation for the box. Conversely, failure occurs if either the `watcher` declares termination when the box is not close enough to the goal, or the box is rotated so far that the `watcher` can no longer perceive it (the threshold is approximately 70°).

The first experiment was a control, while the other two demonstrate fault-tolerance. In the control, two `pushers` had to move the box along a straight-line path approximately 3 meters (the length of our 90%lab). In the second experiment, after the two pushers moved the box approximately 1.2 meters, we simulated a failure by seizing one `pusher` and shutting it off. As a result, the remaining `pusher` was left to push the box by itself, alternating sides under the direction of the `watcher`. In the third experiment we thoroughly tested the system's fault tolerance capabilities. We first let the two pushers push for approximately 0.6 meters, then seized one `pusher` to simulate failure. After the remaining `pusher` had single-handedly pushed the box another 1.2 meters or so, we reintroduced the failed `pusher`, at

[1]Video footage of all of these experiments is available at: `http://fnord.usc.edu/~gerkey/videos`.

| Experiment | $\mu$ | $\sigma$ |
|---|---|---|
| No failure | 31.22 | 0.44 |
| Pusher failure | 132.75 | 26.94 |
| Pusher failure & recovery | 116.44 | 37.72 |

Table 3: *Mean ($\mu$) and standard deviation ($\sigma$) of the elapsed time (in seconds) for the successful pushing trials in each of the three experiments.*

which point they had to finish the task together.

We performed 10 trials each of the above experiments 1-3. There were a total of three failures, one occurring in each set. Two were due to over-rotation of the box, and the third to premature termination on the part of the `watcher`, presumably because of sensor noise. With 27 successes in 30 trials, the two-sided 90% confidence interval for the overall success rate of the system is: $p \in (0.81, 0.99)$

We also analyzed the time elapsed during the successful trials, as a measure of relative efficiency among the different experiments. The results are shown in Table 3. Importantly, the elapsed time for the rather complicated failure and recovery case is less than for the simple failure case. This indicates that any overhead incurred by the reintegration of the recovered `pusher` is outweighed by the benefit of having the two `pushers` cooperating. As for the standard deviation, the monotonic increase is intuitive, since, as the complexity of the situation grows, the exact behavior of the system quickly becomes less repeatable.

In the third experiment, the two `pushers` switched sides when appropriate, which was in half the trials. The appropriateness of switching was determined by the configuration of the box and the remaining `pusher` at the time of re-introduction, and this configuration was in turn a result of the complex system dynamics mentioned above. However, the fact that the pushers automatically switched sides at the right times, with no detriment to the performance, demonstrates that our task-allocation system performs as specified. Details of this work are found in [13, 9].

# 6 Studying Commitment vs. Opportunism

One of the important differences between BLE and Murdoch, above, lies in the amount and duration of commitment a robot has for a given task. To address the impact of commitment-based vs. opportunistic task allocation, we implemented and studied different alternatives in the context of the "emergency handling" experimental problem domain.

## 6.1 The Problem Domain: Emergency Handling

Consider a habitat on the Martian surface whose physical integrity needs to be constantly monitored and maintained for purposes of life-support. In the case of a breach, an alarm is sounded indicating the need for immediate action by one or more of the maintenance robots. The task of these emergency handling robots is to patrol the environment, and upon detecting the sound of an alarm, to use the right tools and take appropriate action to remedy the breach. The emergency handling problem domain provides room for exploration of the key parameters involved in task allocation. For example, by varying the number of robots and alarms, a spectrum of multi-robot coordination problems is generated. Further, by varying other key parameters, such as the availability of local communication, the availability of global communication, the rate at which alarms appear, the time needed to fix alarms in relation to the speed of the robots, etc., a large space of interesting problems is defined.

In all cases, deciding which robot should go where and when can be viewed as a *scheduling problem*. Since the robots do not know in advance when the alarms are going to go off, the scheduling problem is *dynamic*. Furthermore, the task allocation system must be robust to failure of individuals or subsets of the robots, which suggests a distributed approach. Thus we seek a *dynamic, distributed, robust scheduling algorithm* that performs well in all of the above cases, as the parameters are varied.

If multiple robots hear an alarm, deciding which robot should go and fix it, is a key problem. The solution depends on the robots' ability to handle the alarm, their metric distance to the alarm, the interference along the route, the density of other robots in the area, the level of confidence as to where the alarm is, the state of each robot, the presence of other alarms, and so on. To realistically simulate the property that a robot might hear an alarm before it sees the emergency, we developed sound-emitting alarms detectable by the robots' microphones before the robots are within visual range of the sound source. This arrangement enforces locality, in that sound emanating from the source of the alarm generates a gradient. Sound is a convenient synthetic alarm, however the gradient generated is far from being perfectly smooth. Sound reflectivity is a complex property further complicated by a lack of detailed knowledge of the environment. Our work shows that even without a model of the environment (e.g., a map), the overall gradient can be effectively used for effective task allocation.[2]

In the context of the emergency handling task, commitment means that, once assigned, a robot stays dedicated to handling a particular alarm, until the alarm can no longer be detected (i.e., by being out of range or having been put out). Opportunism, on the other hand, means that a robot can switch tasks, i.e., alarms, if for example it detects another alarm with greater intensity or priority. In our experiments, coordination is linked to communication, and the robots are able to communicate information about task allocation, i.e., which robot(s) should service which alarm(s). This is in contrast to individualism, where robots have no awareness of each other, and are uncoordinated. In a coordinated version of task allocation, we used communication to prevent multiple robots from trying to fix the same alarm; robots mutually excluded each other from engaging in the same alarm. This has the important effect of reducing interference among robots, and prevents loss of coverage that might arise if many robots rush to fix a single distant alarm.

### 6.2 A Comparative Study

We designed four experiments to represent the four combinations resulting from studying the two key parameters, coordination vs. individualism, and commitment v. opportunism.

Each emergency handling robot runs the same control program, a hierarchy of behavior modules. Output from an `Emergency Handling` module is fused with output from an `Obstacle avoidance` module, to produce safe navigation in the environment. The `Emergency Handling` module arbitrates between running an `Idle` behavior and running a `Fix Alarm` behavior, based on whether the robot is currently engaged in fixing an alarm. The `Idle` behavior causes the robot to move forward slowly. Combined with `Obstacle avoidance`, the `Idle` behavior causes the robot to move around in the environment at a pace that conserves battery power and minimizes the risk of collision. We gave each alarm a brightly colored cover, which could be detected by the robots' vision system when the robot was very close to the alarm.

The `Fix Alarm` behavior does the following:

1. The robot makes a 360° scan of the environment, monitoring the intensities of the frequency corresponding to its current engagement, and moves in the direction of highest intensity.
2. The robot moves forwards until a junction is detected, then it repeats step 1 above.
3. If the robot detects an alarm with its camera, it servos on vision instead of sound.
4. If the robot is close enough to the alarm, it puts out the sound by emitting a counter-sound at the right frequency.

[2] Clearly sound is only relevant to environments with air, but the results of our work are general and independent of the alarm modality.

| | Individ. | | Mut. Exclusion | |
|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| Commitment | 1955 | 148 | 2084 | 213 |
| Opportunism | 1311 | 531 | 1314 | 433 |

Table 4: *Mean and variance of the total alarm on-time in each experiment. Alarms remaining have not been taken into consideration*

The blackboard algorithm, discussed earlier, runs as a separate thread on the robots. We studied the case of the emergency handling task with a set of three homogeneous robots and a set of four alarms. To ensure homogeneity, each robot is capable of fixing any alarm. The environment chosen was a bounded part of our office environment, including three offices and a copier room (each room is approximately 3m X 4m). The connecting corridor was approximately 15m in length and 2m in width.

Each experiment was performed three times, with the exception of the mutual exclusion/opportunism case for which 4 trials were performed. Mean values and variances of the sum of alarm on-time for each trial are shown in Table 4. Statistical analysis shows that there is a significant difference between the commitment row and the opportunism row, and that the difference between the individualism column and the mutual exclusion column is insignificant. Thus the presence of communication in our scenario plays a negligible role. Details of this work can be found in [16]. We discuss the lessons learned from this work in the next section.

## 7 Discussion and Conclusions

We have described three approaches to dynamic task allocation for groups of robots. We first discussed Broadcast of Local Eligibility (BLE) as a general tool for coordination between robots. BLE extends the *port-arbitrated behavior* (PAB) paradigm across networks of robots. The PAB approach allows a clean, uniform interface between system components (behaviors) at all levels that abstracts away many issues of timing and communication; the "black boxes" of behaviors may contain reactive mappings or deliberative planners. We have validated our BLE approach through experiments in the domain of cooperative multi-robot observation of multiple moving targets, or CMOMMT. We have demonstrated BLE's ability to dynamically assign robots to prioritized tasks through favorable comparisons with a number of other control strategies. BLE allows dynamically-determined group leaders to modify the controllers of other group members on-the-fly. Finally, we demonstrated BLE's ability to assign tasks appropriately to heterogeneous robots.

Next, we described the design and implementation of Murdoch, a market-based auction system for multi-robot coordination, and applied it to box-pushing by teams of mobile robots. Based on the negotiation-style task-allocation facilities provided by MURDOCH, the resulting system is tolerant to robot failures, and efficient in its use of resources. We have demonstrated this system through a series of experiments on a group of Pioneer mobile robots. The results from these experiments are encouraging, and we are currently exploring other directions of this work, including different control laws, better task metrics, and more robots (both `pushers` and `watchers`).

Finally, our experiments with the emergency handling task, clearly show that the opportunistic strategy worked significantly better that the commitment-based strategy. We suggest that this might be because the time taken by a single robot to reach the source of an alarm was significantly larger than the time it took a robot to fix an alarm, once a robot was there. This choice of the parameter space favors opportunism over commitment since the former effectively uses the presence of robots near emergencies by harnessing them immediately. In other regions of the parameter space of the emergency handling task (e.g., where the ratio of time-to-reach-alarm to the time-to-fix-problem is small) opportunism might not be as effective. The present study excluded the case where several robots would be required to fix an alarm in a cooperative fashion. This is also a regime where performance might improve with commitment. Exploring the variations in the parameter space in order to derive significant general trends in this problem of emergency handling is a rich area for future research. Finally, our results do not show any performance improvement by using communication in our version of the emergency-handling task. This does not mean, of course, that communication is not extremely useful in various distributed coordination problems (such as those addressed by BLE and Murdoch, both of which are based on communication for coordination), but it is interesting that it does not have a helpful effect in this particular instance. To test the generality of this result we plan in the future to experiment with the cooperative case requiring multiple robots to fix an alarm.

In summary, the approaches discussed in this paper share a common theme: each addresses an instance of the general problem of multi-robot task allocation for multiple goals. Our results were obtained in the context of three NASA-relevant applications (multiple target tracking, object manipulation, and emergency handling) which were used to experimentally validate our algorithms.

## Acknowledgments

This paper is a review of research at the USC Robotics Research Laboratories, supervised by the authors, and conducted by Brian Gerkey (Murdoch), Esben Ostergaard (blackboard) and Barry Werger (BLE). Portions of these results were published previously.

This work is supported in part by DARPA grant DABT63-99-1-0015 under the Mobile Autonomous Robot Software (MARS) program, ONR equipment grant N00014-00-1-0638, and NSF grants ANI-9979457 and ANI-0082498 under the Special Projects in Networking Program.

## References

[1] Sample return and other missions. http://mars.jpl.nasa.gov/missions/future/2005-plus.html.

[2] Ronald C. Arkin. Cooperation without communication: Multiagent schema based robot navigation. *Journal of Robotic Systems*, 9(3), April 1992.

[3] Ronald C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

[4] Tucker Balch and Ron Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 1999.

[5] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, April 1986.

[6] Daniel D. Corkill. Blackboard systems. *AI Expert*, 6(9):40–47, September 1991.

[7] B. Gerkey and M.J. Mataric. Murdoch: Publish/subscribe task allocation for heterogeneous agents. In *Proceedings of Autonomous Agents 2000*, pages 203–204, Barcelona, Spain, June 3–7 2000.

[8] Brian Gerkey and Maja J Matarić. Principled communication for dynamic multi-robot task allocation. In *International Symposium on Experimental Robotics 2000, Waikiki, Hawaii, December 10-13*, 2000.

[9] Brian P. Gerkey and Maja J Matarić. Murdoch: Publish/subscribe task allocation for heterogeneous agents. In *Proceedings, The Fourth International Conference on Autonomous Agents (Agents 2000)*, pages 203–204, Barcelona, Spain, June 2000.

[10] T.L. Huntsberger, P. Pirjanian, and P.S. Schenker. Robotic outposts as precursors to a manned mars habitat. In *Proc. Space Technology and Applications International Forum (STAIF-2001)*, Albuquerque, NM, February 2001.

[11] Maja J Matarić. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2–4):321–331, December 1995.

[12] Maja J. Matarić. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2–3):323–336, 1997.

[13] Maja J Matarić and Brian P. Gerkey. Principled communication for dynamic multi-robot task allocation. In *Proceedings, International Symposium on Experimental Robotics*, pages 341–352, Waikiki, Hawaii, 2000.

[14] A. Mishkin, J. Morrison, T. Nguyen, H. Stone, B. Cooper, and B. Wilcox. Experiences with operations and autonomy of the mars pathfinder microrover. In *1998 IEEE Aerospace Conference Proceedings*, pages 337–351, 1998.

[15] E. Ostergaard, Maja J. Mataric, and Gaurav S. Sukhatme. Distributed multi-robot task allocation for emergency handling. In *2001 IEEE/RSJ IROS (submitted)*, 2001.

[16] E. Ostergaard, Maja J. Mataric, and Gaurav S. Sukhatme. Distributed multi-robot task allocation for emergency handling. Technical Report IRIS-01-402, Institute for Robotics and Intelligent Systems, University of Southern California, 2001.

[17] L. Parker. Alliance: An architecture for fault-tolerant multi-robot cooperation. In *IEEE Transactions on Robotics and Automation, 14(2):220–240*, 1998.

[18] Lynne E. Parker. Cooperative motion control for multi-target observation. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1591–1598, 1997.

[19] Lynne E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing, special issue on Robotics Research at Oak Ridge National Laboratory*, 5(1):5–19, 1999.

[20] P. Pirjanian, T.L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, H. Das, S. Joshi, and P.S. Schenker. Campout: A control architecture for multi-robot planetary outposts. In *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III*, volume 4196, Boston, MA, November 2000.

[21] P.S. Schenker, T.L. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. Das, S. Joshi, H. Aghazarian, A.J. Ganino, B.A. Kennedy, and M.S. Garett. Robot work crews for planetary outposts: Close cooperation and coordination of multiple robots. In *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III*, volume 4196, Boston, MA, November 2000.

[22] B. Werger and M. Mataric. Broadcast of local eligibility for multi-target observation. In *Proceedings, of DARS.*, Oct. 2000.

[23] Barry Brian Werger and Maja J Matarić. Broadcast of local eligibility for multi-target observation. In *Proceedings, 5th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 347–356, Knoxville, TN, 2000.

[24] Barry Brian Werger and Maja J Matarić. From insect to internet: Situated control for networked robot teams. *Journal of Annals of Mathematics and Artificial Intelligence*, to appear, 2001.